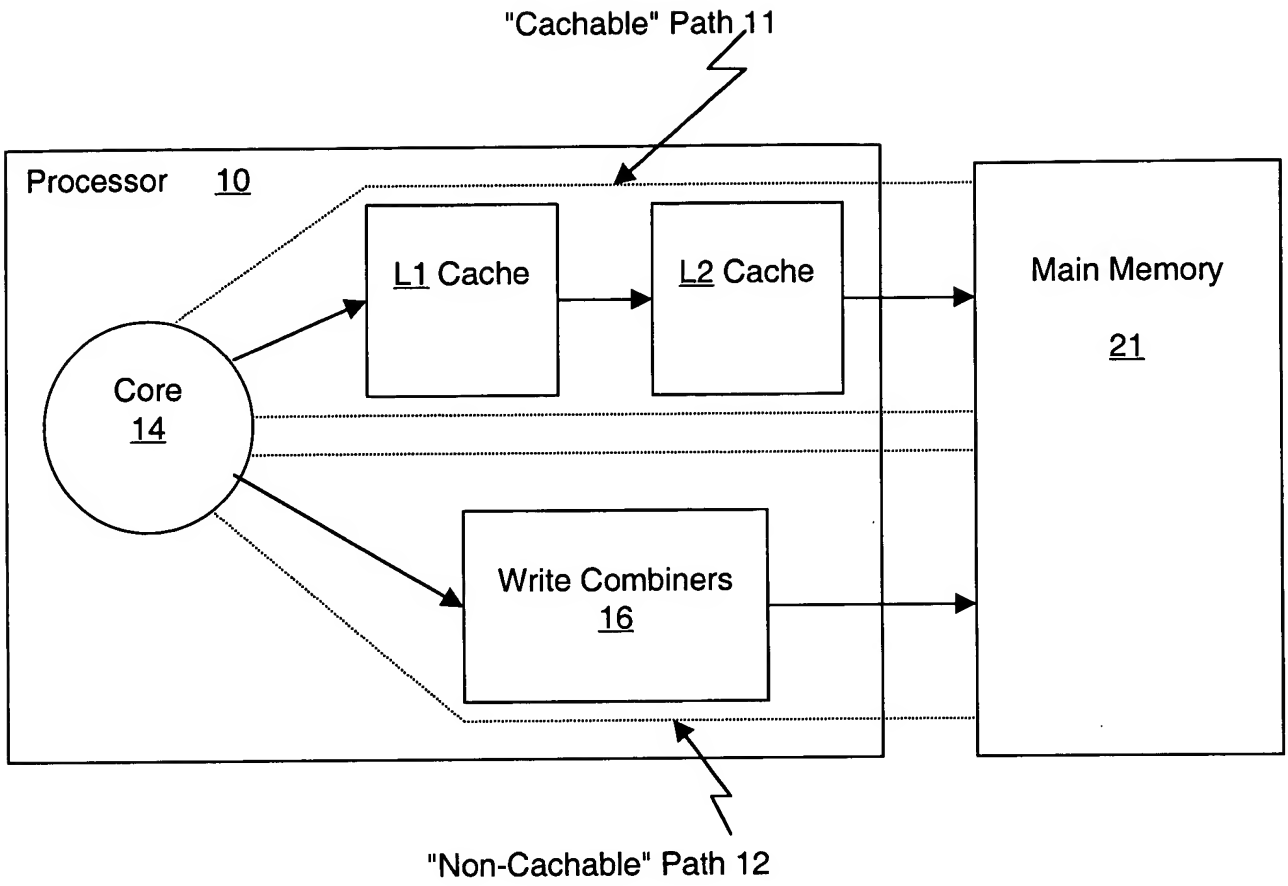
"Cachable" Path 11
Processor 10
L1 Cache
L2 Cache
Main Memory
21
Core
14
Write Combiners
16
"Non-Cachable" Path 12

Fig. 1A

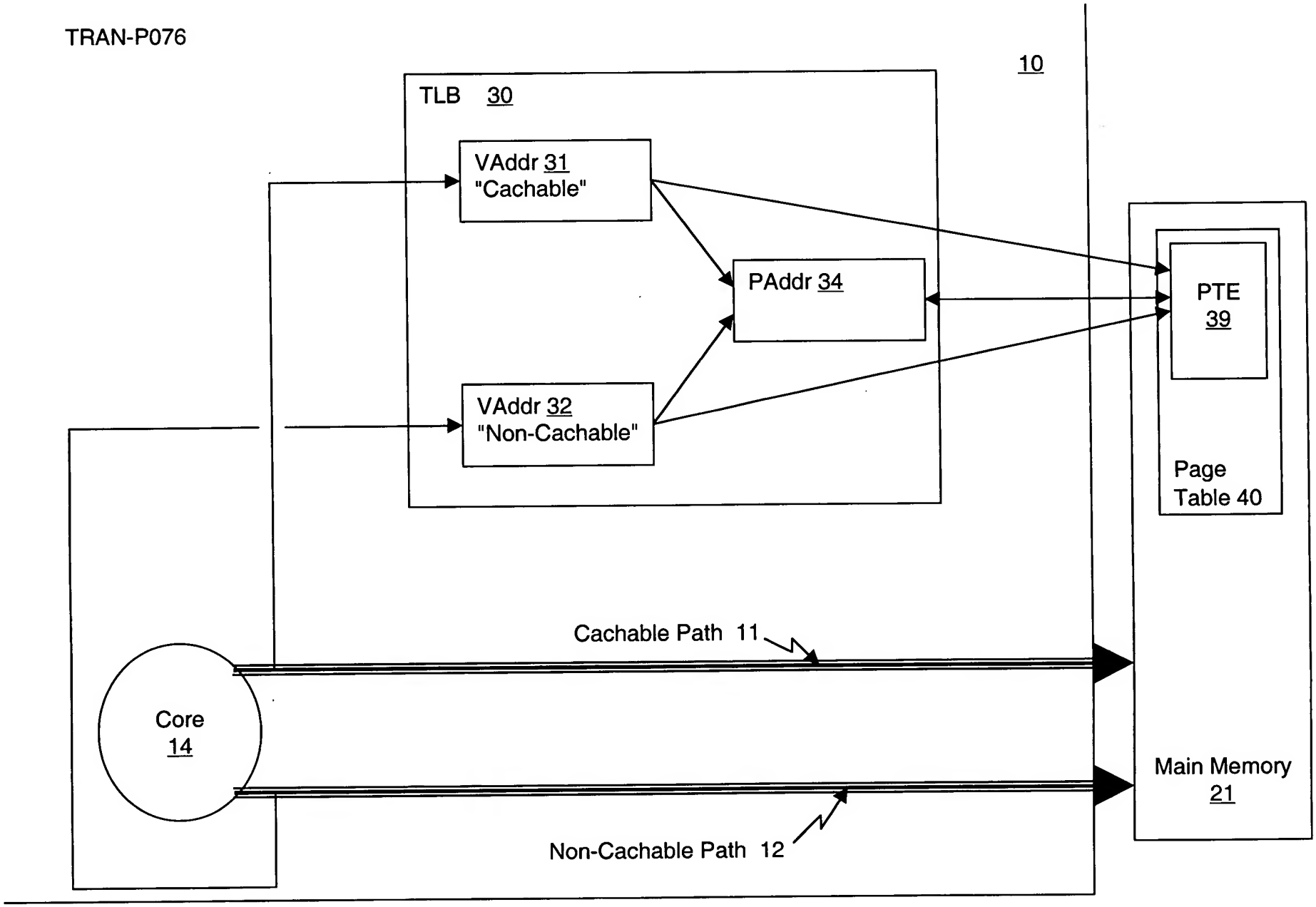
10
TLB 30
VAddr 31
"Cachable"
PAddr 34
VAddr 32
"Non-Cachable"
PTE
39
Page
Table 40
Cachable Path 11
Core
14
Main Memory
21
Non-Cachable Path 12

Fig. 1B

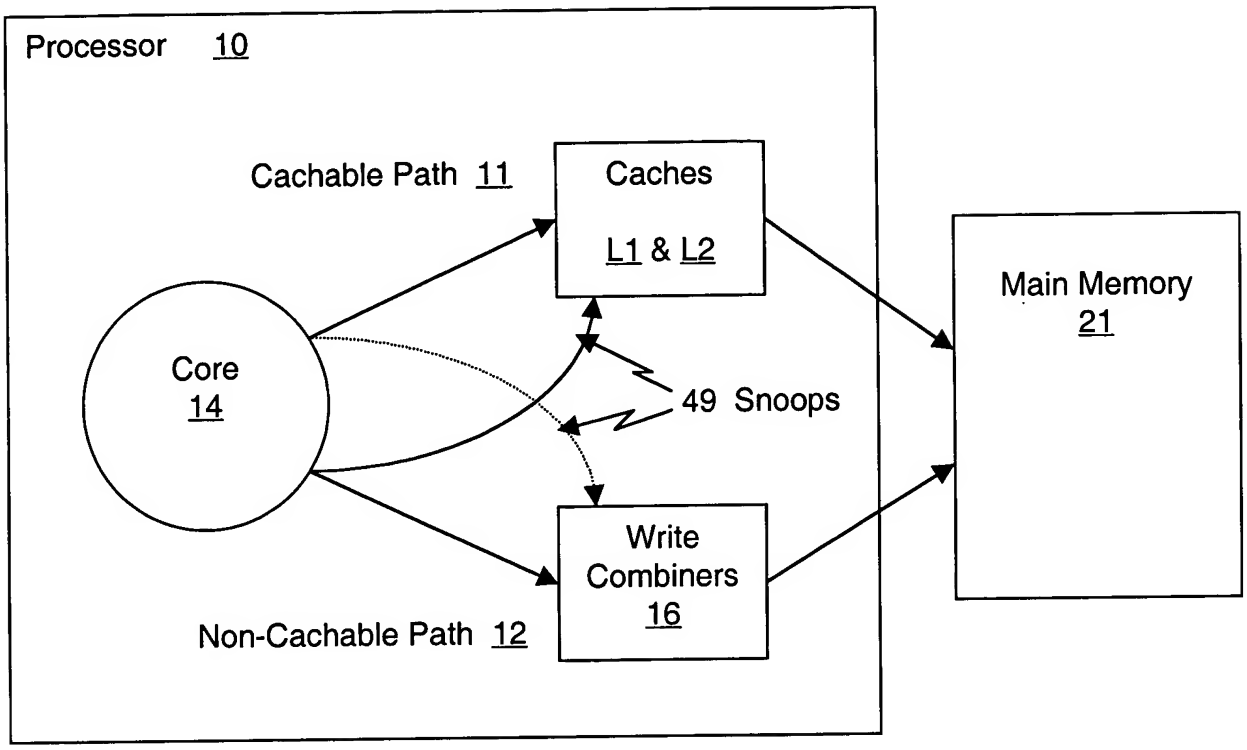
Processor 10
Cachable Path 11
Caches
L1 & L2
Main Memory
21
Core
14
49 Snoops
Write
Combiners
16
Non-Cachable Path 12

Fig. 1C

200

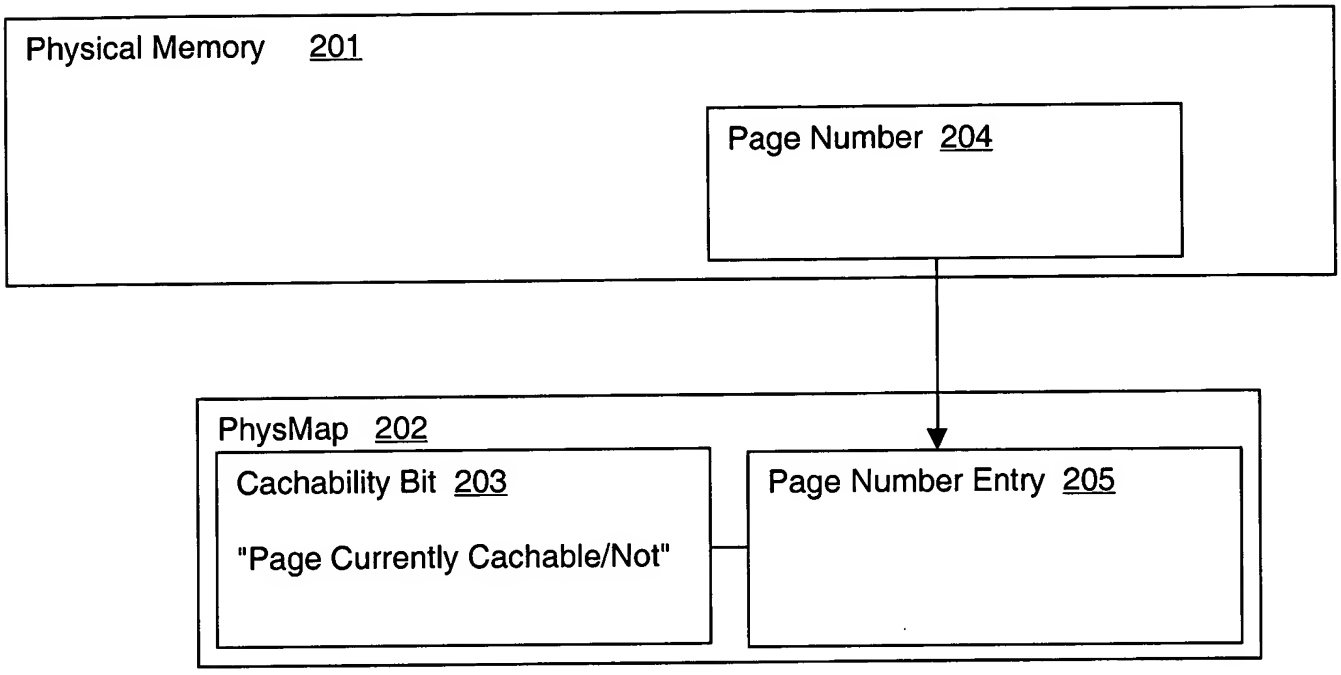

Fig. 2

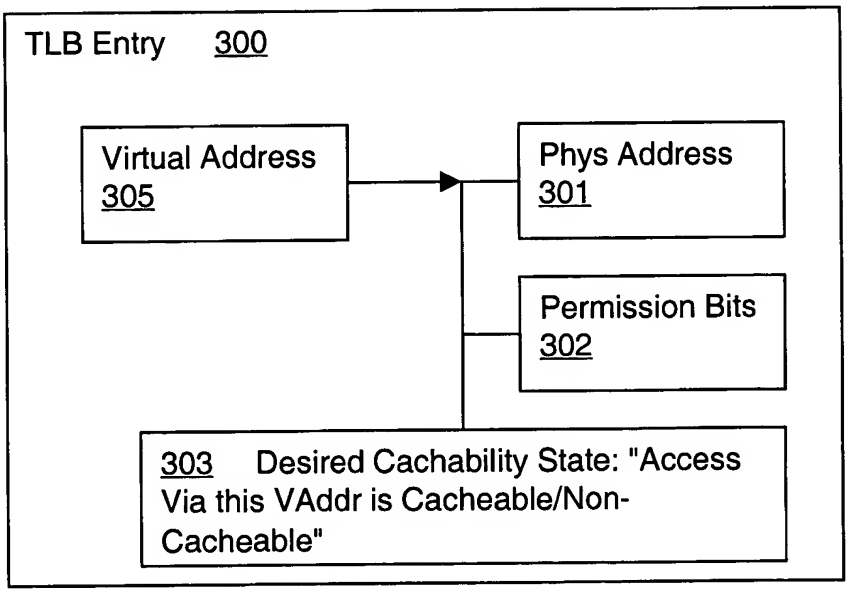

Fig. 3

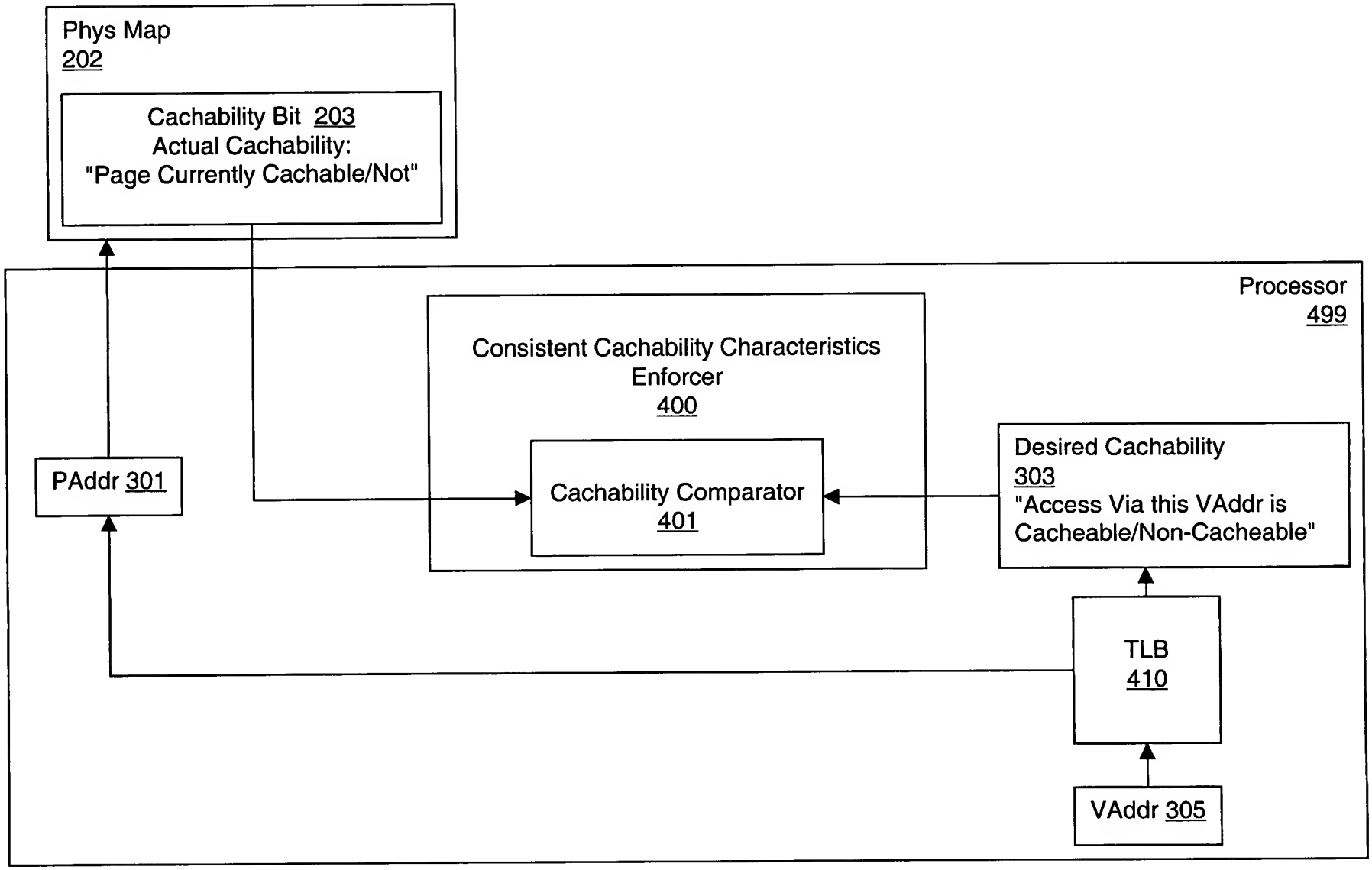
Phys Map
202
Cachability Bit 203
Actual Cachability:
"Page Currently Cachable/Not"
Processor
499
Consistent Cachability Characteristics
Enforcer
400
Cachability Comparator
401
PAddr 301
Desired Cachability
303
"Access Via this VAddr is
Cacheable/Non-Cacheable"
TLB
410
VAddr 305

Fig. 4

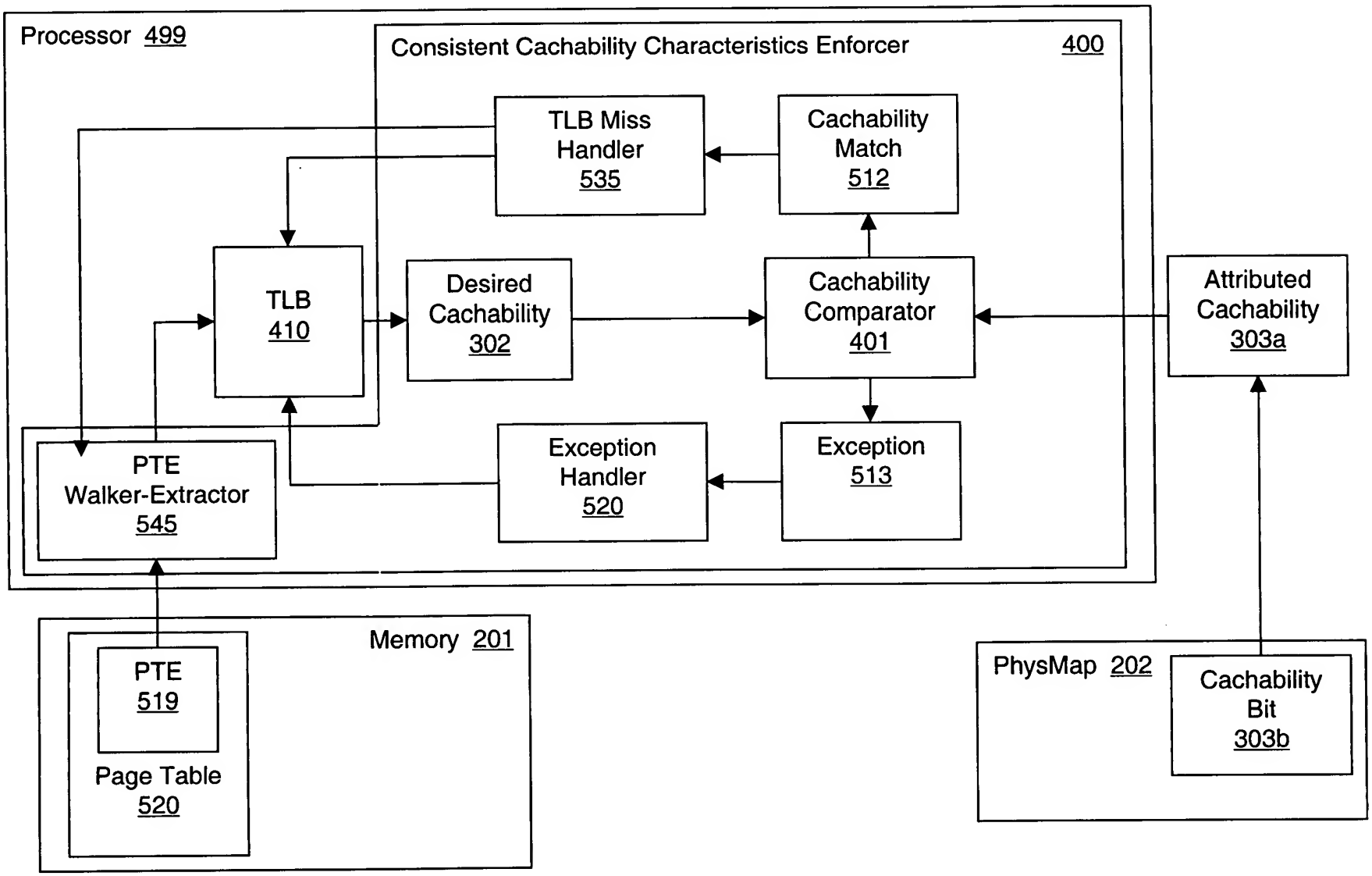
Processor 499
Consistent Cachability Characteristics Enforcer 400
TLB Miss Handler 535
Cachability Match 512
TLB 410
Desired Cachability 302
Cachability Comparator 401
Attributed Cachability 303a
PTE Walker-Extractor 545
Exception Handler 520
Exception 513
Memory 201
PTE 519
Page Table 520
PhysMap 202
Cachability Bit 303b

Fig. 5

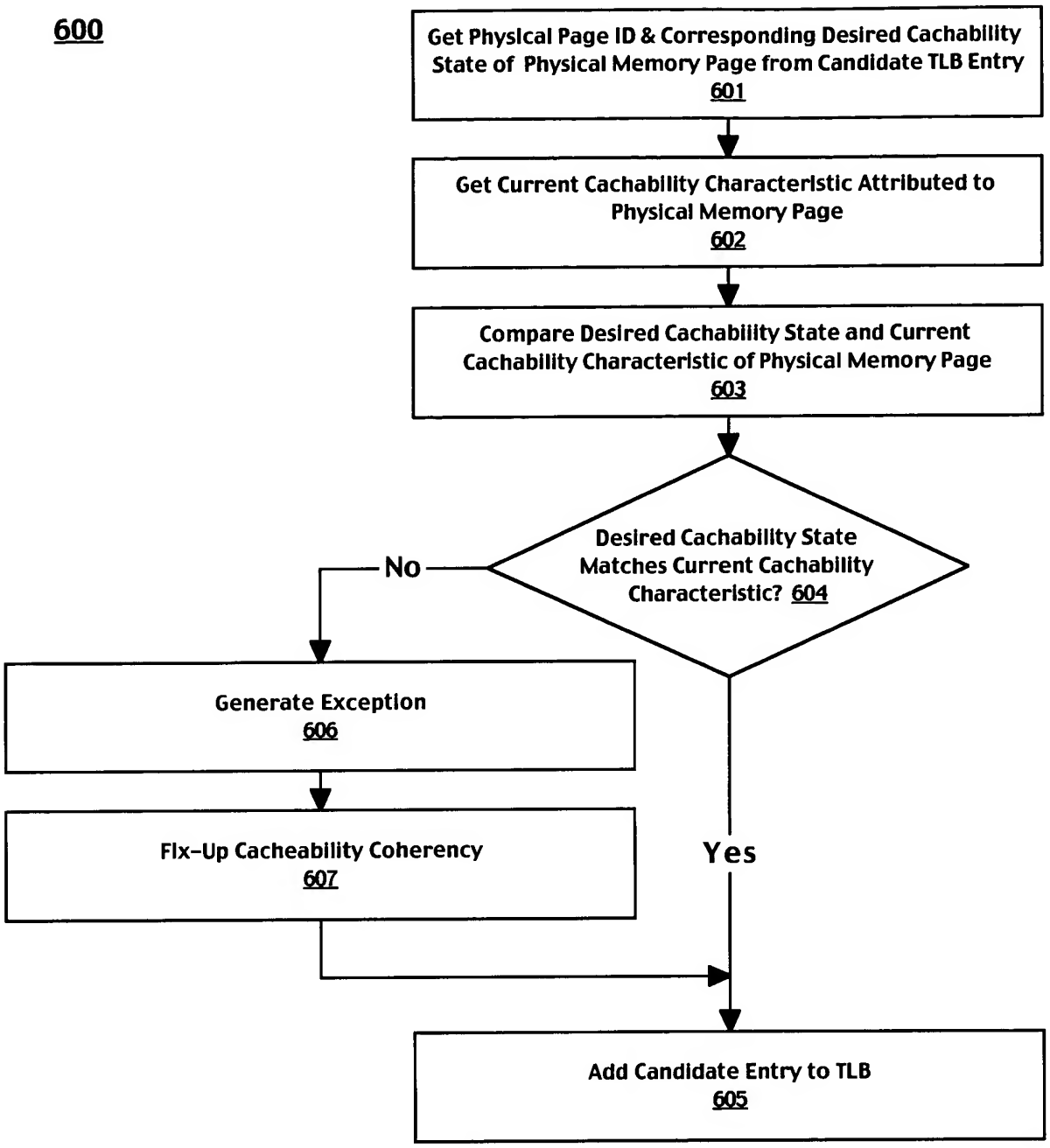
600
Get Physical Page ID & Corresponding Desired Cachability State of Physical Memory Page from Candidate TLB Entry
601
Get Current Cachability Characteristic Attributed to Physical Memory Page
602
Compare Desired Cachability State and Current Cachability Characteristic of Physical Memory Page
603
Desired Cachability State Matches Current Cachability Characteristic? 604
No
Yes
Generate Exception
606
Fix-Up Cacheability Coherency
607
Add Candidate Entry to TLB
605

Fig. 6

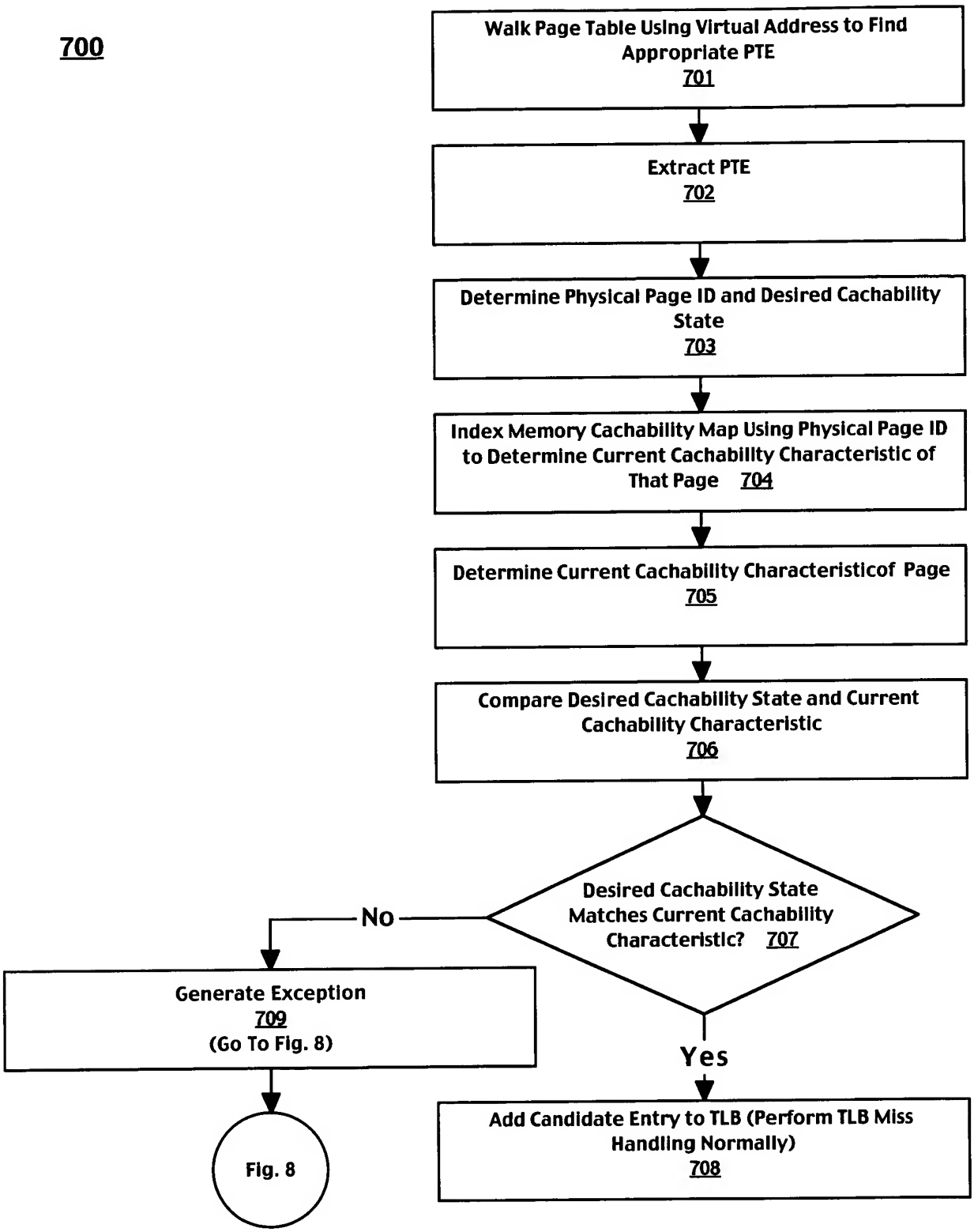
700
Walk Page Table Using Virtual Address to Find Appropriate PTE
701
Extract PTE
702
Determine Physical Page ID and Desired Cachability State
703
Index Memory Cachability Map Using Physical Page ID to Determine Current Cachability Characteristic of That Page 704
Determine Current Cachability Characteristicof Page
705
Compare Desired Cachability State and Current Cachability Characteristic
706
Desired Cachability State Matches Current Cachability Characteristic? 707
No
Yes
Generate Exception
709
(Go To Fig. 8)
Fig. 8
Add Candidate Entry to TLB (Perform TLB Miss Handling Normally)
708

Fig. 7

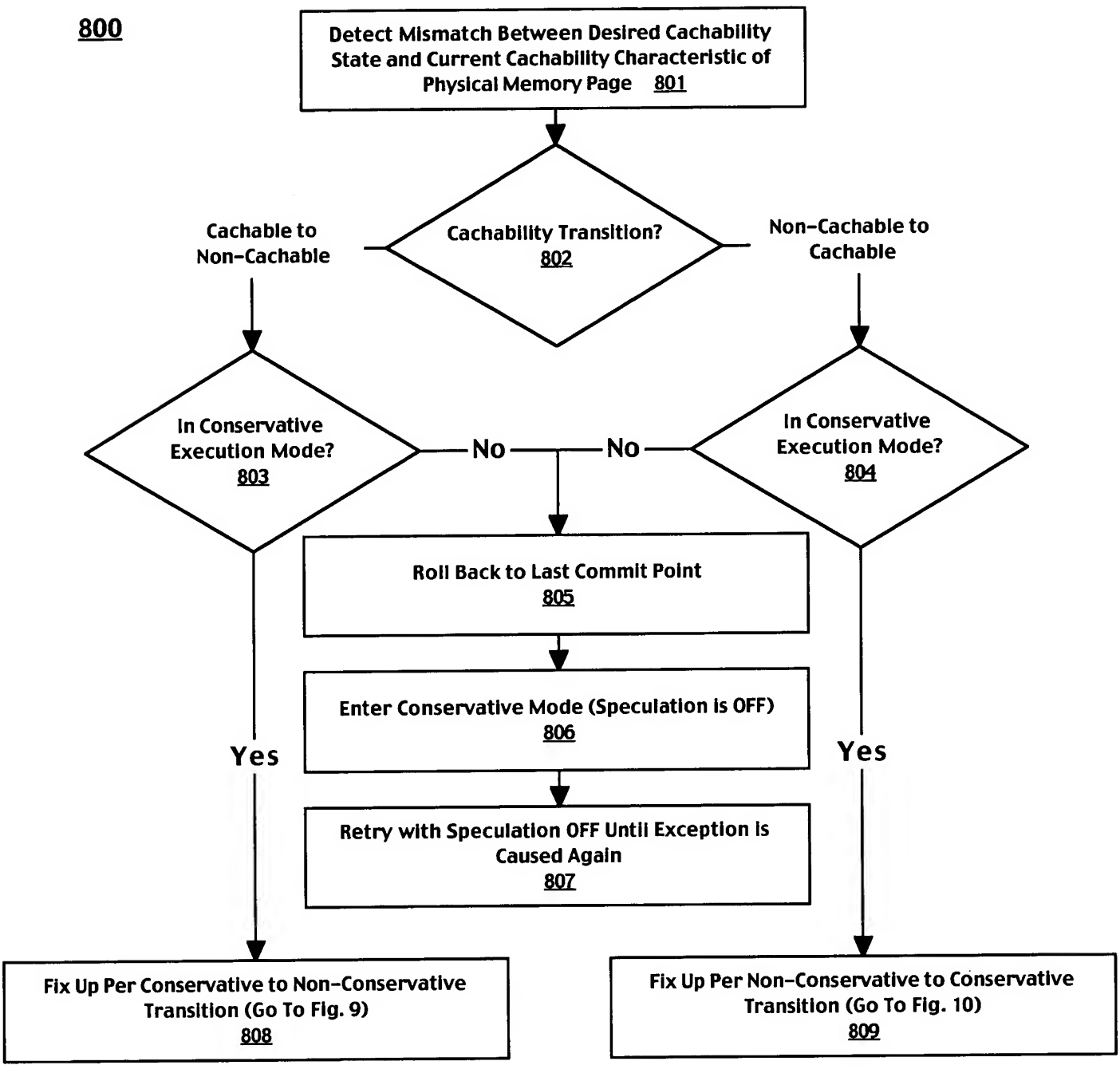
800
Detect Mismatch Between Desired Cachability State and Current Cachability Characteristic of Physical Memory Page 801
Cachability Transition? 802
Cachable to Non-Cachable
Non-Cachable to Cachable
In Conservative Execution Mode? 803
In Conservative Execution Mode? 804
No
No
Yes
Yes
Roll Back to Last Commit Point 805
Enter Conservative Mode (Speculation is OFF) 806
Retry with Speculation OFF Until Exception is Caused Again 807
Fix Up Per Conservative to Non-Conservative Transition (Go To Fig. 9) 808
Fix Up Per Non-Conservative to Conservative Transition (Go To Fig. 10) 809

Fig. 8

900

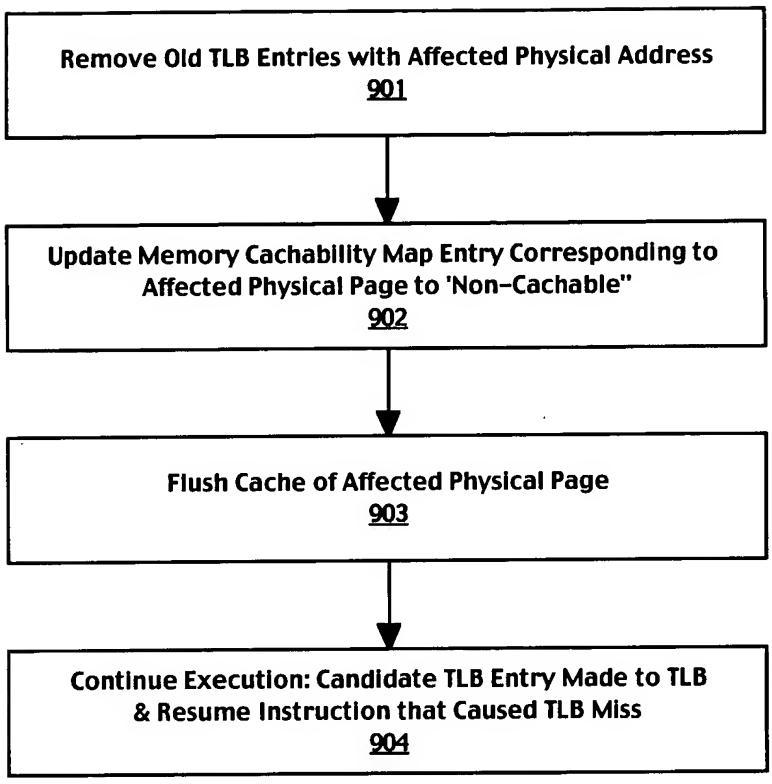
Remove Old TLB Entries with Affected Physical Address
901
Update Memory Cachability Map Entry Corresponding to Affected Physical Page to 'Non-Cachable"
902
Flush Cache of Affected Physical Page
903
Continue Execution: Candidate TLB Entry Made to TLB & Resume Instruction that Caused TLB Miss
904

Fig. 9

1000

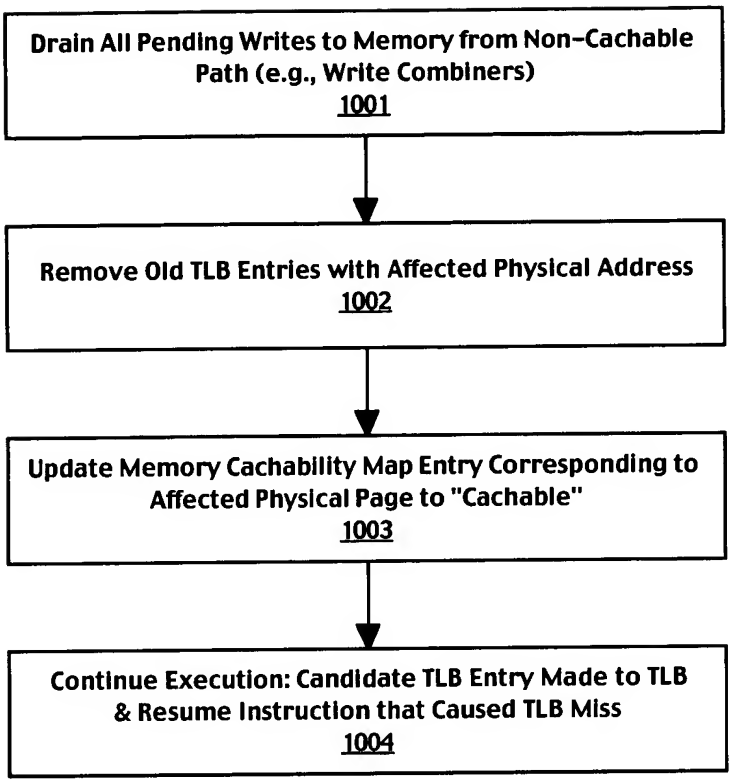
Drain All Pending Writes to Memory from Non-Cachable Path (e.g., Write Combiners)
1001
Remove Old TLB Entries with Affected Physical Address
1002
Update Memory Cachability Map Entry Corresponding to Affected Physical Page to "Cachable"
1003
Continue Execution: Candidate TLB Entry Made to TLB & Resume Instruction that Caused TLB Miss
1004

Fig. 10